

Microsoft SMB – Looking Forward

Tom Talpey
Microsoft



Outline

- A look at SMB3 today
- A look at things “in the works” in Windows
- The SMB1 situation
- Other uses of SMB3



SMB3 Today

- SMB3 is the key storage protocol for Windows Interoperability
 - Export Win32 file API over network
 - Extensible to other semantics, and multiple transports
- SMB3 is a mature protocol
 - Shipped since 2012, on SMB2 foundation
 - On diverse platforms – Microsoft, Samba, and many others
- Many other uses
 - Hyper-V, Storage Spaces Direct, Windows Clustering, etc



SMB3-related Features in Development

- SMB Direct Push Mode
- Signing optimization/improvement
- QUIC
- Compression
- Client-driven write through
- SMB Server move on connect
- Identity Tunneling for Hyper-V
- SMB Global Mounts for containers



SMB3 Signing

- GMAC (Galois Message Authentication) support
- Discussed at Prior SDC
 - Computationally highly efficient
 - Lightweight alternative to full AES CMAC
- Considering support in Windows



SMB3 over QUIC

- QUIC is lightweight new transport over UDP
 - Advanced connection, congestion, etc efficiency
 - Increasing deployment for HTTP, other upper layers
- Considering for SMB3
- See Mathew's upcoming SDC talk



SMB3 Compression

- SMB3 TRANSFORM_HEADER can support compression as transform
 - With definition of a new ProtocolID field
 - Currently, only 0xFD,'S','M','B' == encryption
 - Other transforms possible
- Interesting for both local and remote
 - Especially, optimizing common transfers e.g. all-0
- See Mathew's upcoming SDC presentation



SMB3 Client-driven Writethrough

- SMB3.1.1 already defines per-operation writethrough
 - SMB2_WRITEFLAG_WRITE_THROUGH
- Additional use by future Windows client
 - In support of Hyper-V, filesystem-over-SMB use



SMB3 Server Move-On-Connect

- SMB3.1.1 already supports TreeConnect Context
 - Extended Error redirects recent clients to optimal server
- Upcoming use by Windows ScaleOut File Server (SOFS)
 - To optimize clients without use of optional SMB Witness protocol
 - Locally implemented optimal server “forwarding” still supported



SMB3 Identity Tunneling, Global Mount

- As discussed in previous SDC talks
- Identity Tunneling allows “system” identity for authorized access
 - Useful for Hyper-V scenarios, where trust already established
- Global Mount allows shares to be automatically mounted
 - Useful for container scenarios, where lightweight contexts come and go
- Implementation and additional use by future Windows client



Two SMB3 Things from Last Year's SambaXP

SMB Direct Linux - Review

- Microsoft prototyping SMB Direct support for Linux!
- Client-only
 - Not contemplating doing a server implementation
- Simplifying initial principles:
 - Connections are made via RDMA directly – no TCP, no multichannel
 - Basic RDMA transfers only



SMB Direct – last year

- Initial implementation
 - Connects, and negotiates SMB3.1.1 on Windows Server RDMA connection
 - Transfers data successfully
 - Currently, fails on sustained file copy (server detects MID out-of-range)
- Not (yet) supported:
 - SMB Direct placement (RDMA Read/Write)
 - Requires explicit memory registration and care with RDMA verbs (completions)
 - Full multichannel, with fallback/forward
 - Requires significant client transport architecture work



SMB Direct – this year

- Full implementation
 - Connects, and negotiates SMB3.1.1 on Windows Server RDMA connection
 - Transfers data inline and via RDMA
- Working on:
 - Direct i/o from user buffers and uncached
- Long Li's talk later in conference



Unix (Posix) Extensions – last year

- The key to Linux SMB3 support
- And, long-overdue
- Do we now have sufficient understanding of requirements?
- Of a protocol?
- It's time to move forward.



Unix (Posix) Extensions – this year

- Samba Server implementation (Samba Team)
- Samba Client in development (Microsoft, Samba Team)
- Support for Linux/Unix/Posix guests envisioned in Azure cloud
- Steve French's talk later in conference



SMB1 Must Die

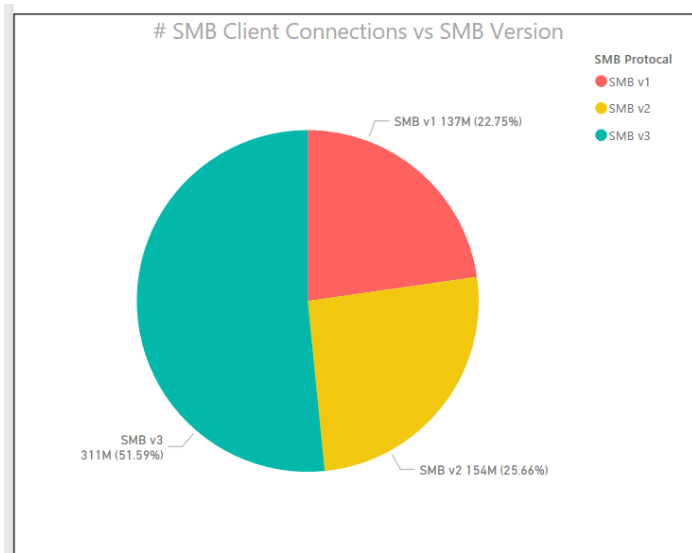
Still Needs SMB1

- Surely you all know of <https://aka.ms/stillneedssmb1>
- The clearinghouse has been growing (new software and devices identified) but also shrinking (vendors updating to support SMB2 and 3).
- A meaningful one this month: Ubuntu Linux stopped defaulting to SMB1 support in 17 and updated 18 LTS (BIONIC BEAVER) release to avoid use by default:
 - https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes#Default_CIFS.2FSMB_protocol_version_change_in_CIFS_mounts.
- Linux ecosystem to follow?



Windows 10 SMB1 Telemetry – June 5 2018

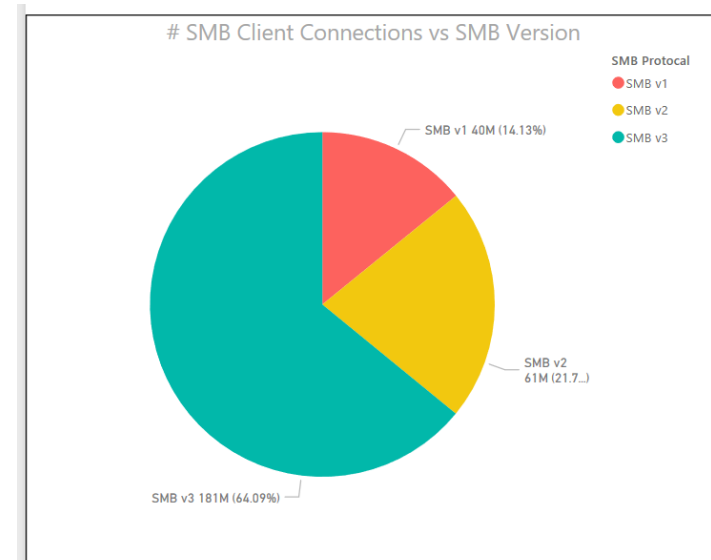
Windows 10 Professional – down to 23% from 45% 1yr



Total number of devices reporting telemetry for this event for last 7 days

1.45M

Windows 10 Enterprise – down to 14% from 24% 1yr



Total number of devices reporting telemetry for this event for last 7 days

341.38K

SMB1 Servers

- Windows Servers largely switching off SMB1
- Samba Servers still supporting SMB1
- Especially, remaining 3.x servers
 - Unable to upgrade?
 - Unwilling to upgrade?
 - Stuck on 3.x (GPL2) codebase?
- Need to un-stick this!
- Outright drop SMB1 support in future Samba 4.x?
- Force 3.x servers to move forward?



Durability

Tom's annual suggestion for future engagement



Windows PMEM Support

- Persistent Memory is supported in Windows 10 and Windows Server 2016
 - PM support is foundational in Windows and is SKU-independent
- Support for JEDEC-defined NVDIMM-N devices available in
 - Windows Server 2016
 - Windows 10 (Anniversary Update – Fall 2016)
- Access methods:
 - ✓ Direct Access (DAX) Filesystem
 - Mapped files with load/store/flush paradigm
 - Cached and noncached with read/write paradigm
 - ✓ Block-mode (“persistent ramdisk”)
 - Raw disk paradigm
 - ✓ Application interfaces
 - Mapped and traditional file
 - NVM Programming Library
 - “PMEM-aware” open coded



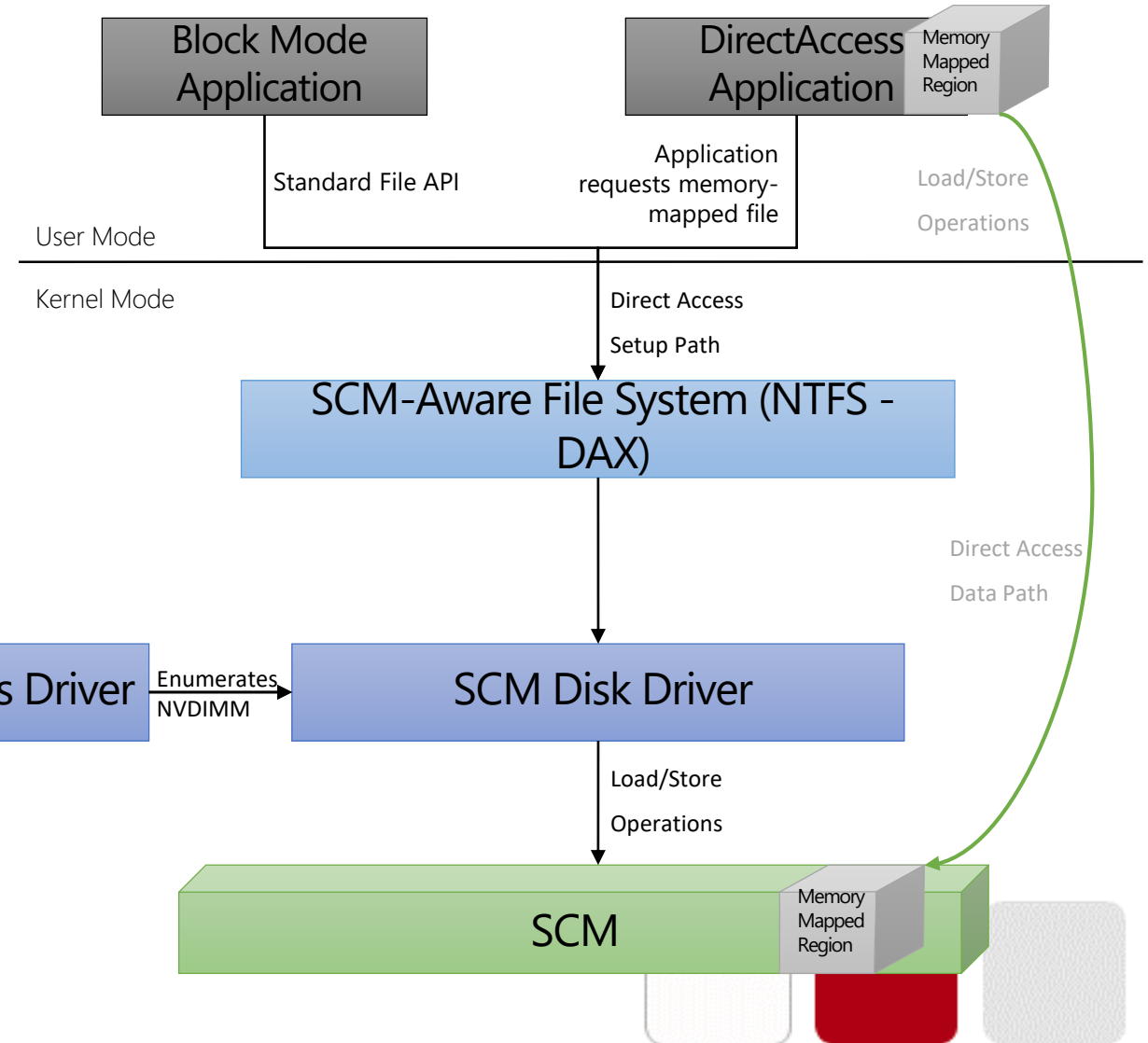
Direct Access Architecture (Windows)

Overview

- Support in Windows Server 2016 and Windows 10 Anniversary Update (Fall 2016)
- App has direct access to Storage Class Memory (SCM/Pmem) via existing memory-mapping semantics
- Updates directly modify SCM, Storage Stack not involved
- DAX volumes identified through new flag

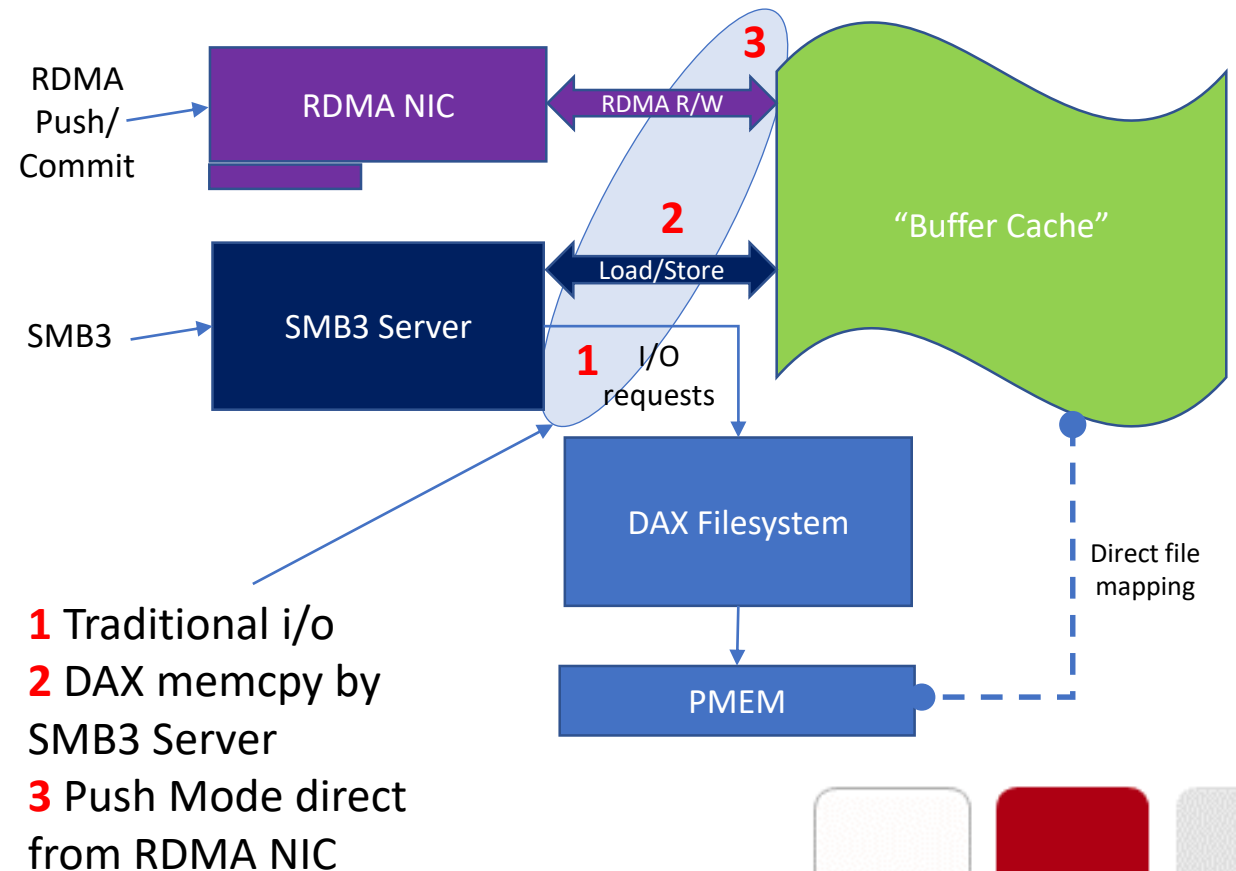
Characteristics

- True device performance (no software overhead)
- Byte-Addressable
- Filter Drivers relying on I/O may not work or attach – no I/O, new volume flag
- AV Filters can still operate (Windows Defender already updated)



Going Remote – SMB3

- SMB3 RDMA and “Push Mode” discussed at previous SNIA Storage Developers Conferences
- Enables zero-copy remote read/write to DAX file
 - Ultra-low latency and overhead
- 2, 3 can be enabled even *before* RDMA Commit extensions become available, with slight extra cost



RDMA Protocol Extensions

- Two extensions advancing in IBTA (IB, RoCE)
- RDMA Flush is flush to durability
- Atomic Write places pointer-sized data after flush
 - Transactional, e.g. for log write pointer update
- IETF (iWARP) discussion also active

- Push Mode only envisions using RDMA Flush



Windows Push Mode Investigation

- Previously, Phase 1 discussed at SDC
- Currently, Phase 2 under investigation
 - See Mathew's talk at SDC this coming September
- Phase 3 also possible
 - With prototype RDMA NIC implementation



Push Mode in Samba Server?

- Push Mode is quite convenient for Server to implement
- Needs only an RDMA connection, and a long-lived memory registration
 - And a region of DAX/PMEM, of course 😊
- Server CPU not needed for data transfer
- SMB3 protocol not extended
 - Only a new FSCTL
- Interested?



THANK YOU

